CSCE 475H
Final Project Assignment:
Local Decisions vs. Global Coherence

Assigned: February 10, 2015

Due: April 30, 2015

(Project 5 minutes late will *not* be accepted)

Introduction

The goal of this final project is to learn how to design a multiagent system (MAS) where agents make local decisions leading to coherent emergent behaviors that satisfy the design objectives. Through this exercise, you will be able to learn to appreciate the complexity of such a design as well as the simplicity of the resultant design once it is developed. The key is that we know that in general more global directives and more global-centric decisions will lead to better coherence more easily than very local-centric decisions and very little global control. However, a MAS with more global directives and more global-centric decisions reduce the autonomy and flexibility the individual agents. When agent autonomy is reduced, the unique benefits of having an agent-based solution are also reduced. This has been one of the most important problems that MAS researchers have tried to work ever since the inception of the MAS research area.

In this final project assignment, you will submit a proposal for your final project, use the Repast simulation software, run comprehensive experiments, and report on your design and results.

* *If a team intends to conduct a study different from the "local decisions vs. global coherence" objective, I am open to that but I do require the team to argue and justify their choice, to make a compelling case.*

Repast Simulation Software

Check out http://repast.sourceforge.net/index.html for a detailed description of the testbed. There is also FAQ at the website – please review them carefully. Briefly, as reported on the website: The Recursive Porous Agent Simulation Toolkit (Repast) is one of several agent modeling toolkits that are available. Repast borrows many concepts from the Swarm agent-based modeling toolkit.

In particular, check out Repast Simphony at http://repast.sourceforge.net/.

- Repast Simphony 2.2, *released on 26 June 2014*, is a tightly integrated, richly interactive, cross platform Java-based modeling system that runs under Microsoft Windows, Apple Mac OS X, and Linux. It supports the development of extremely flexible models of interacting agents for use on workstations and small computing clusters.
- Repast Simphony models can be developed in several different forms including the ReLogo dialect of Logo, point-and-click statecharts, Groovy, or Java, all of which can be fluidly interleaved. NetLogo models can also be imported.
- Repast Simphony has been successfully used in many application domains including social science, consumer products, supply chains, possible future hydrogen infrastructures, and

ancient pedestrian traffic to name a few. Repast offers built-in simulation results logging and graphing tools.
- Repast has automated Monte Carlo simulation framework.
- Repast provides a range of two-dimensional agent environments and visualizations.
- Repast allows users to dynamically access and modify agent properties, agent behavioral equations, and model properties at run time.
- Repast includes libraries for genetic algorithms, neural networks, random number generation, and specialized mathematics.
- Repast includes built-in systems dynamics modeling.
- Repast has social network modeling support tools.

And our Intelligent Agents and Multiagent Systems (IAMAS) Group at UNL also has developed additional reasoning packages such as case-based reasoning, reinforcement learning, MDPs, etc. for use with Repast.

A demonstration of the Repast environment will be given in the classroom on February 10, 2015.

Design and Implementation

You must design an agent that makes local decisions through communicating (directly and/or indirectly) with other agents in the multiagent systems. The goal is to minimize global control, keeping those directives to a minimum, and yet still maintain a coherent, emergent behavior that satisfies the design objectives of the system. Thus, you must be able to articulate the motivations for an agent to make the correct local decisions and how. Then, you must be able to show that if every agent makes the correct local decisions (and they will since they are all motivated to do so), then the desired overall outcomes can be achieved. Since a MAS lives in an environment, you also need to specify the environmental parameters that will in turn help you design your agent.

This is not easy. Thus, you will probably have multiple agent designs, conduct numerous experiments, and iteratively refine your agent design and your environmental factors.

You are required to turn in only one demo-ready agent in **Java** for every type of agent in your environment**.** For example, if you are simulating an auction scenario, your environment would include an auctioneer and a set of buyers. In that case, you need to turn in one auctioneer agent and one buyer agent. Your simulation program then would replicate that buyer agent to generate the required set of buyer agents.

You should also make sure that your Java programs run on **Windows**. For example, if you write your program in Linux, and port it to Windows, you may see some extra weird characters attached to the end of each line of the code. Such characters may prevent the windows environment from running your program correctly.

Experiments and Report

You are required to run an experiment with your agent design. You must propose a set of hypotheses that you want to validate. And then you must design a study that will allow you to collect the data that you need to validate or invalidate the hypotheses. It is likely that you will have many different environmental settings and different agent designs. Systematically evaluate them and report on the results. For your report, you must provide **POJI** of the results:

Presentations, Observations, Justifications, and Implications. We will discuss this further in class.

It is important that your experiments discuss how you achieve coherent, emergent behavior via local decisions. In particular, show the empirical data on communication, coordination, and so on, that support your claim that you minimize global control in local decision making and still achieve coherent, emergent behavior. Of course, this is a "range". And thus, degrees of global control and the quality of the emergent behavior will have to taken into account in your discussion.

Notice that the description of simulation experiments should summarize all of your trial simulation runs. This summary should include: (a) The Simulation Parameters (b) The Initial and Final Values of the Environment Variables (that you want to observe), (c) the Random Number Generator and the Seed (*do not use timestamps, use constant long integers for each run*), (d) Description of the Participating Agents (what type and how many), (e) Summary of the Initial and Final State of Each Agent (e.g., the amount of money an agent has in an auction simulation) (e) Total Tick Counts, and (f) The Final Emergent Behavior. In brief, your simulation summary should allow us to replicate your trial experiments by running your submitted code. Note that your proposal will essentially be a part of your report. So, the better you do your proposal, the less you will have to do towards the end of the semester when you write up your final report.

Final Project Demo Day

On the Final Project Demo day, each team will present their agent and work. Each team is also required to introduce its team before the demo, including the overall strategies. This is to let all students know about the different approaches chosen. With 8 teams and 75 minutes of class time, each team will be allocated about 8 minutes for their entire demo and presentation.

Team

You will be working as a team, the same team that you have already formed (for your Game Days, seminar, and collaborative topic summary). All members of a team will receive the same score unless there is explicit indication given by some members of the team to the instructor (followed with an investigation to determine whether a team member has indeed failed to contribute adequately).

Important Dates

March 2:	A sufficiently detailed proposal that describes (1) the problem statement, (2) the agent design strategy, (3) the desired emergent behavior, (4) the hypotheses, and (5) the experiments that you will likely conduct. No more than 5 pages in length.
April 15:	Last day to submit your final agent design so that we can start running them to collect statistics on local decisions vs. global coherence. (**Note: This gives you at least 15 days to do your experiments and POJI.**)
April 30:	Final Project Report Submission and Demo Day!

Grading

The final project will be graded in 2 parts: programming (50%) and report (50%). The programming part will be graded based on: (a) 45% Program Correctness, (b) 15% Software Design, (c) 10% Programming Style, (d) 15% Testing, and (e) 15% Documentation.

The report will be graded based on: (a) 50% Design Description and Discussion, (b) 20% Organization, (c) 10% Requirements, (d) 10% Description of Simulation Experiments, and (d) 10% Grammar and Errors. The report must be written in a "manuscript format" compliant to AAAI, ACM, or IEEE. These manuscript formats can be found online at:

- AAAI: http://www.aaai.org/Publications/Author/author.php
- ACM: http://www.acm.org/pubs/submissions/submission.htm
- IEEE: http://www.acm.org/pubs/submissions/submission.htm

The quality of the design will be based on our own testing of your software. The TA will collect the following data on your local decisions: (a) Number of Agents, (b) Initial Values of the Environment Variables, (c) Initial and Final State of Each agent, (e.g., the amount of money an agent has in an auction simulation), (d) Total Number of Messages Sent.

And we will collect the following data on the convergence rate of your system and how close the outcome performance matches the expected outcome: (a) Simulation Parameters, (b) Number of Agents, (c) The Final values of the Environment Variables, (d) Emergent Behavior of the System, and (e) Total Tick Counts (i.e., number of timesteps).